
JWORKS
POWERED BY ORDINA
Cloud Practice
EKS workshop
Deploying your application on a Kubernetes cluster
ORDINA
Ahead of change

INTRODUCTION

What to expect from this workshop?

- Kubernetes introduction
- EKS infrastructure explanation
  - Semi-managed implementation by Amazon
- Jenkins jobs
- Demo / workshop
  - Deploy an application to our Kubernetes cluster
  - Set up an automated pipeline which will set this up for you

What is Kubernetes?

- Container-orchestration system
- Open source
  - CNCF (Cloud Native Computing Foundation)
- Deployment
- Scaling
- Failover
- Uses containers (Docker, Cri-O, …)

KUBERNETES

JWorks Kubernetes

- JWorks Development cluster (production ready)
- Hosted at Amazon
  - Elastic Kubernetes Service (EKS)
  - Control + master services managed by Amazon
  - Worker nodes not managed
- Other providers
  - Google (GKE)
  - Azure (AKS)
  - DigitalOcean

Amazon EKS

Google Kubernetes Engine

Azure Kubernetes Service (AKS)

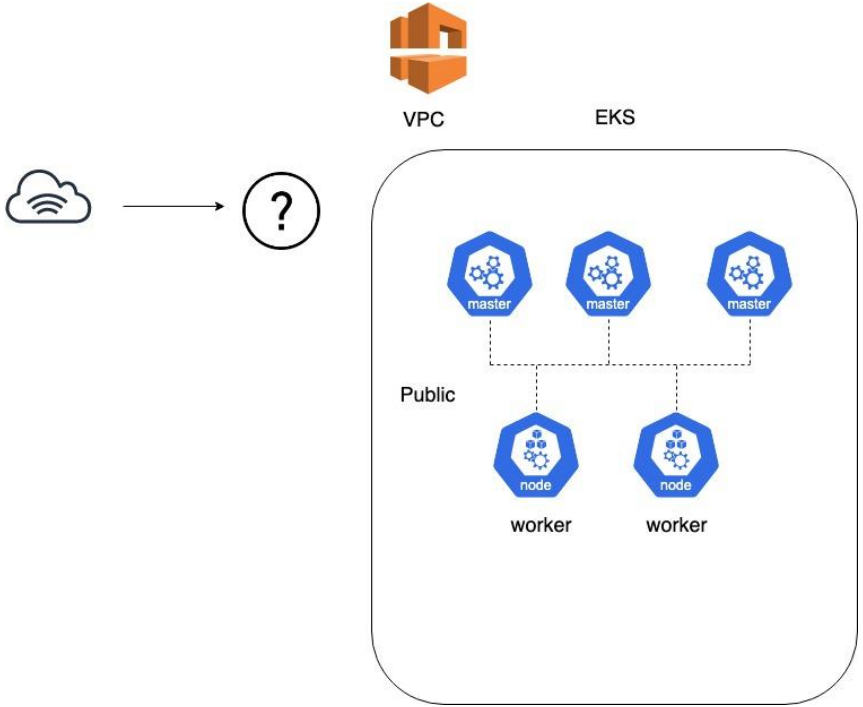
JWORKS

POWERED BY ORDINA

ORDINA

EKS

Kubernetes cluster (default) setup with 3 masters and 2 worker nodes

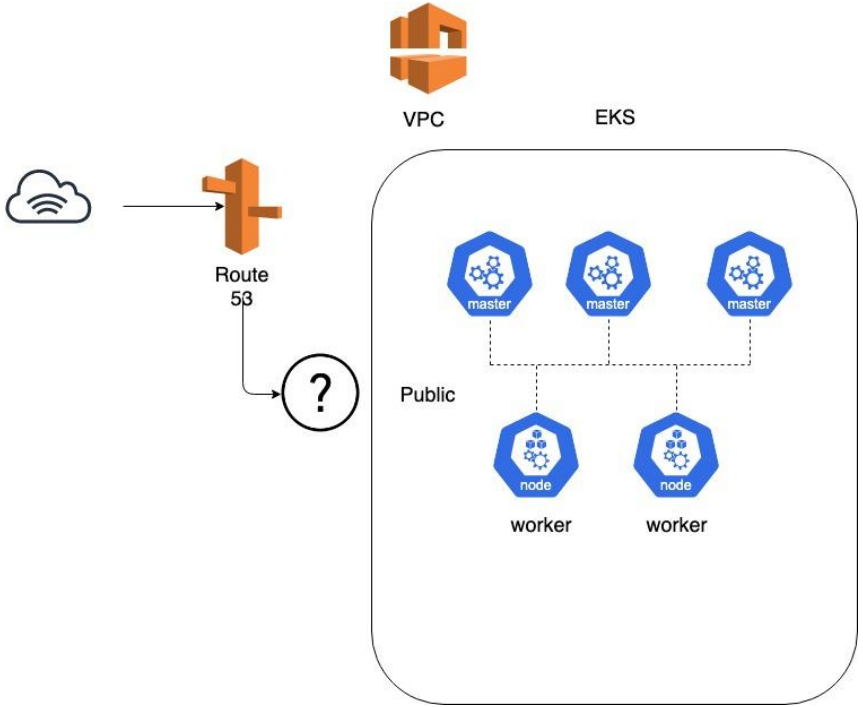


Incoming request - how to reach our Spring Boot application?

EKS

Route 53 hosted zone catches all the requests

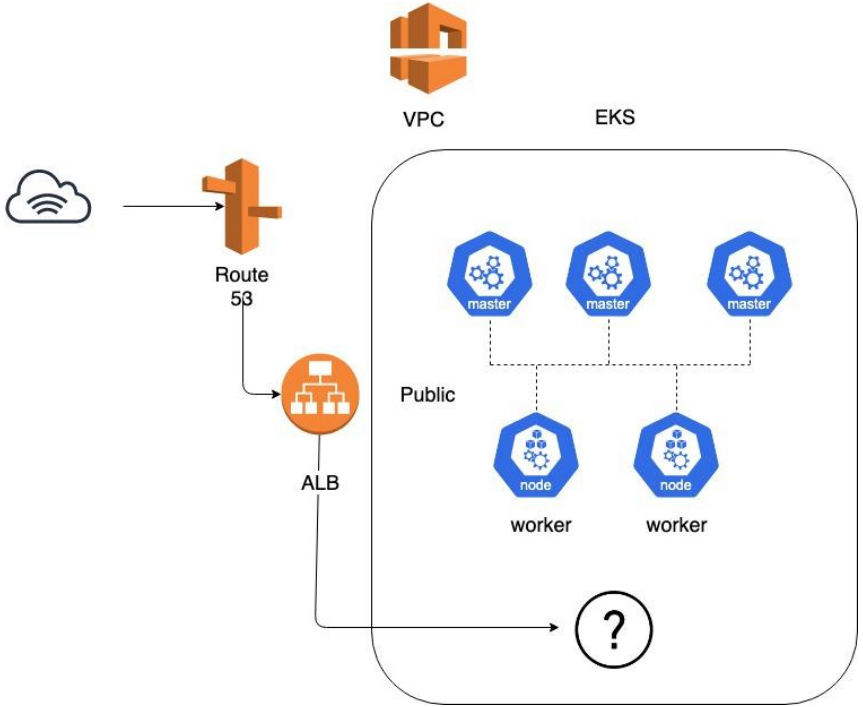
eks.ordina-jworks.io is the root domain for this cluster



EKS

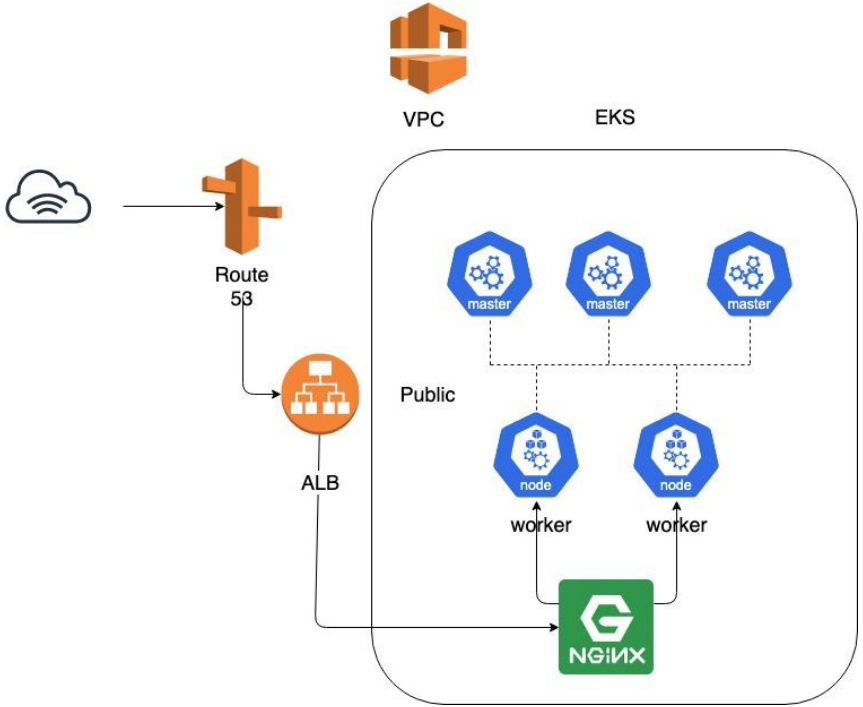
ALB (Application Load Balancer) distributes the request to the respective pod

*ALB matches on the target group



EKS

NGINX sends the request to the appropriate Kubernetes service



AUTHENTICATION

How do we access the cluster?

- The owner of the cluster has full admin access by default
- Since multiple people use our AWS account we probably want to give multiple people access to the cluster as well
- We can use AWS IAM authentication in combination with Kubernetes RBAC to enable kubectl for other people

AUTHENTICATION

AWS IAM

- Users, Groups, Roles
  - A user belongs to a group
  - A group has policies attached
  - A user can assume a Role

Users | Permissions | Access Advisor

Managed Policies

The following managed policies are attached to this group. You can attach up to 10 managed policies.

Attach Policy

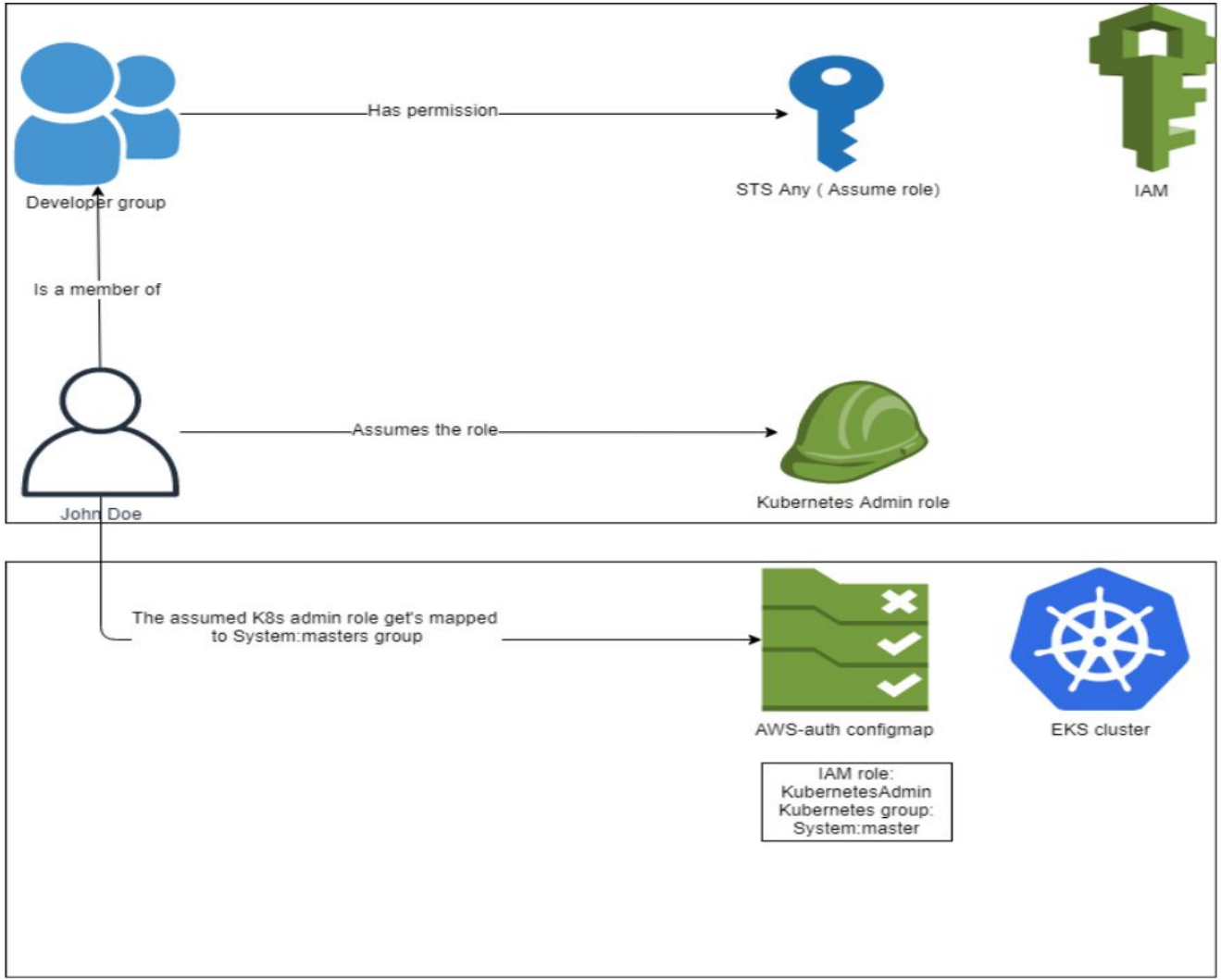
Policy Name	Actions
AmazonAPIGatewayAdministrator	Show Policy | Detach Policy | Simulate Policy
CloudwatchDeveloperPolicy	Show Policy | Detach Policy | Simulate Policy
LambdaDeveloperPolicy	Show Policy | Detach Policy | Simulate Policy
IAMDeveloperPolicy	Show Policy | Detach Policy | Simulate Policy
StsAny	Show Policy | Detach Policy | Simulate Policy

KUBERNETES RBAC

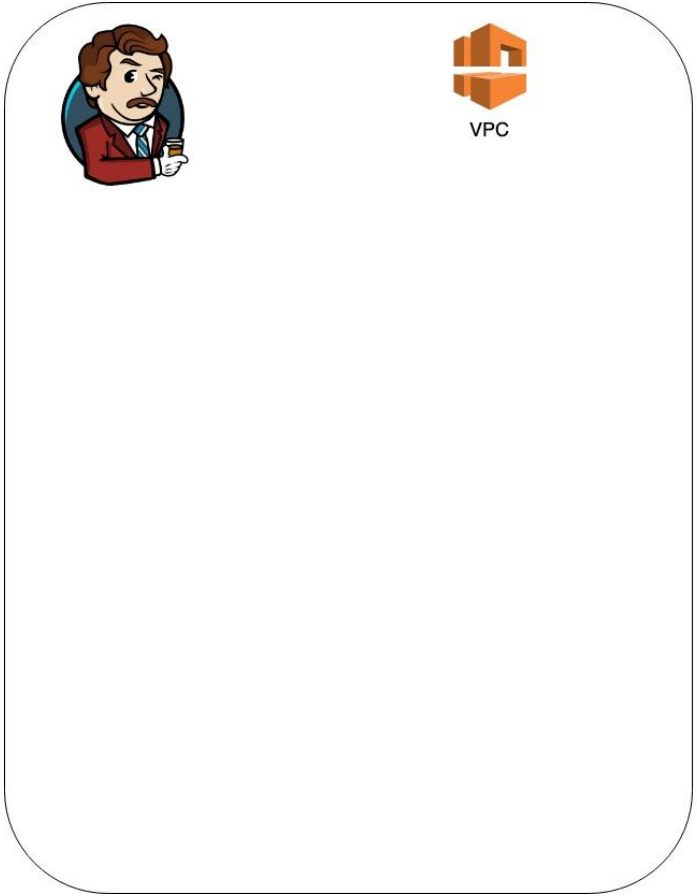
Authentication inside the cluster

- Configmap aws-auth in cluster
- You can bind IAM users / roles to Kubernetes groups.

```
  - rolearn: arn:aws:iam::163091829738:role/KubernetesAdmin
    username: kubernetes-admin
    groups:
      - system:masters
```


Developer group
Has permission
STS Any (Assume role)
IAM
Is a member of
John Doe
Assumes the role
Kubernetes Admin role
The assumed K8s admin role get's mapped
to System:masters group
AWS-auth configmap
EKS cluster
IAM role:
KubernetesAdmin
Kubernetes group:
System:master

JENKINS

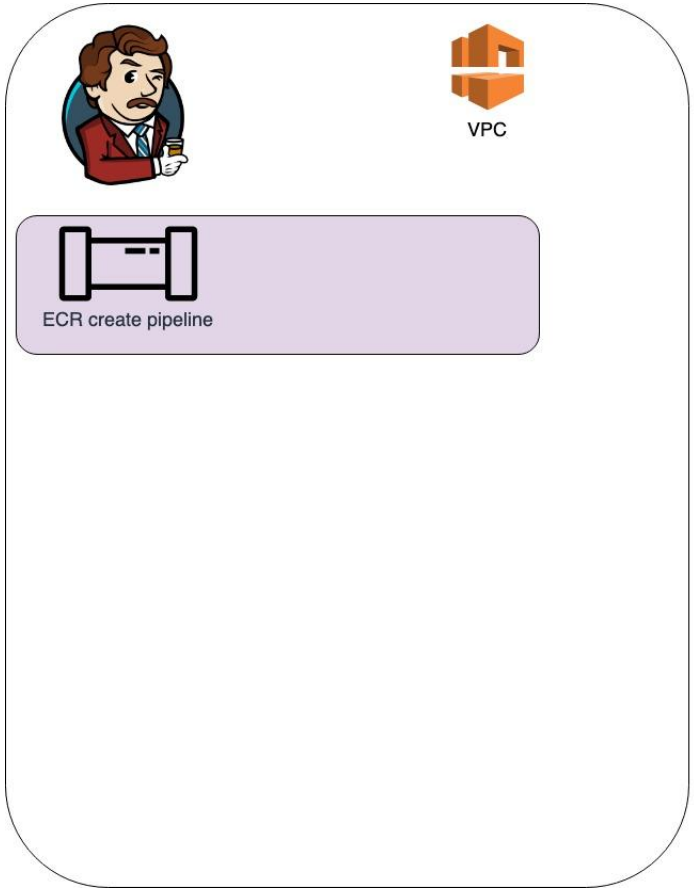


JWORKS

POWERED BY ORDINA

ORDINA

JENKINS

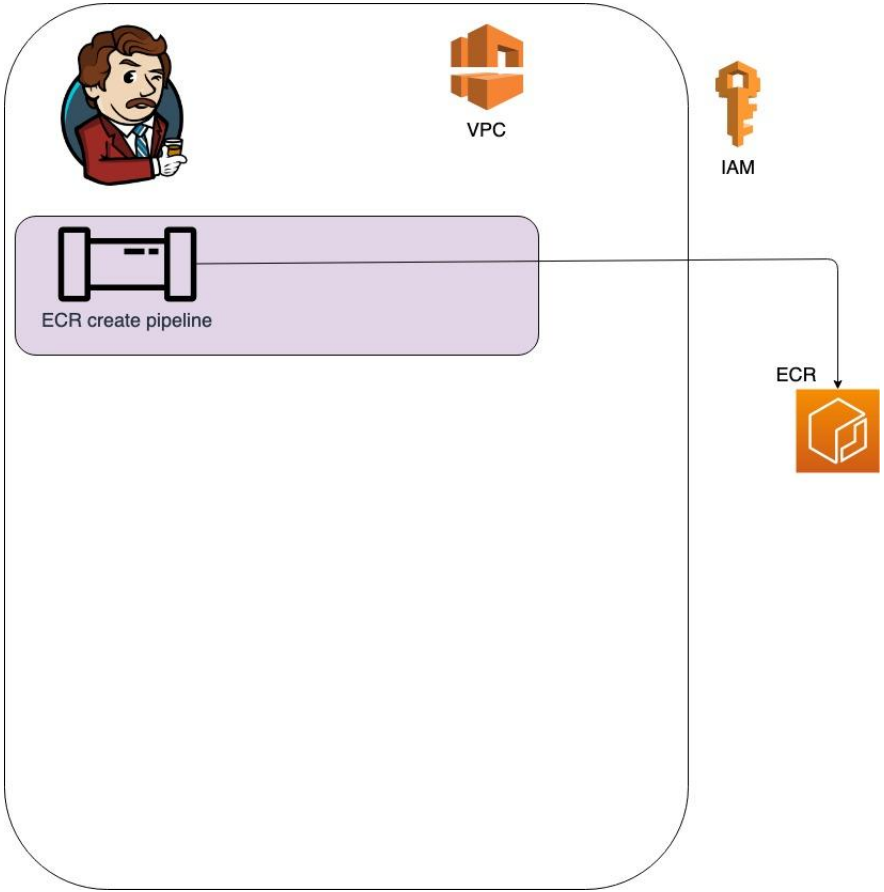


JWORKS

POWERED BY ORDINA

ORDINA

JENKINS

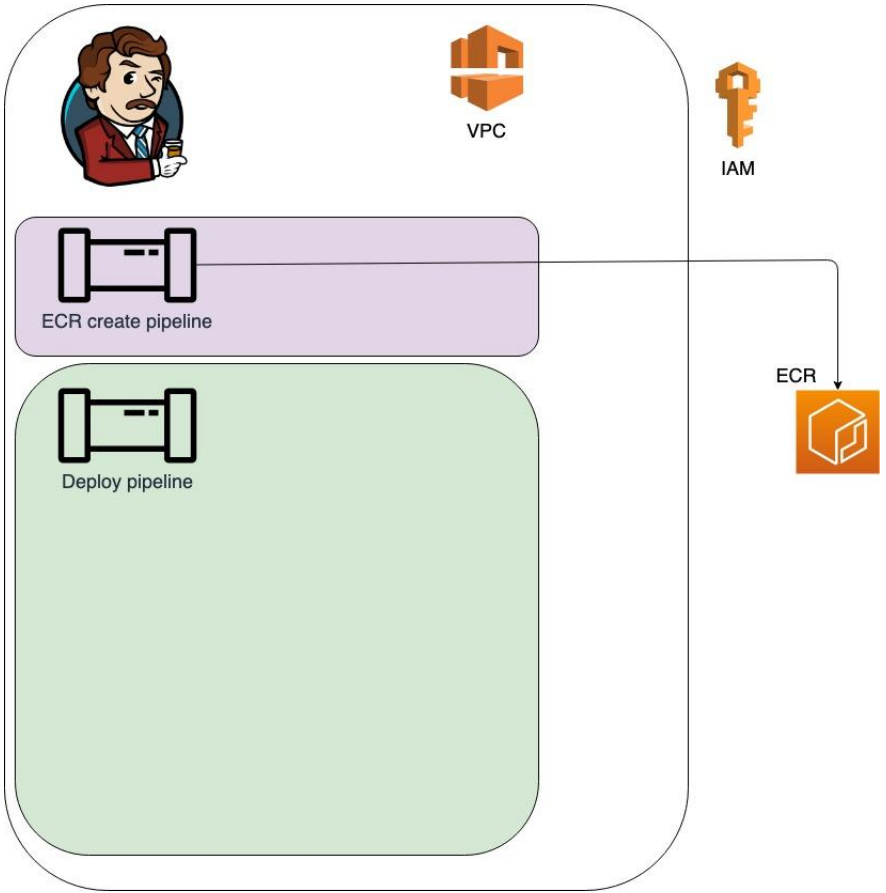


JWORKS

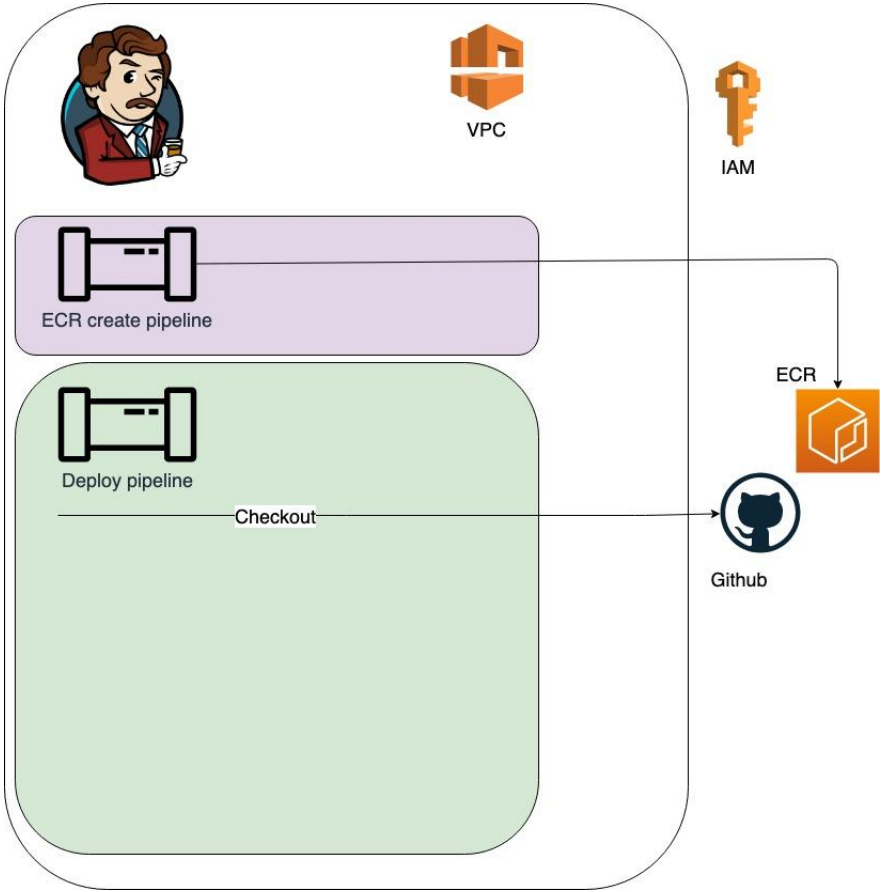
POWERED BY ORDINA

ORDINA

JENKINS



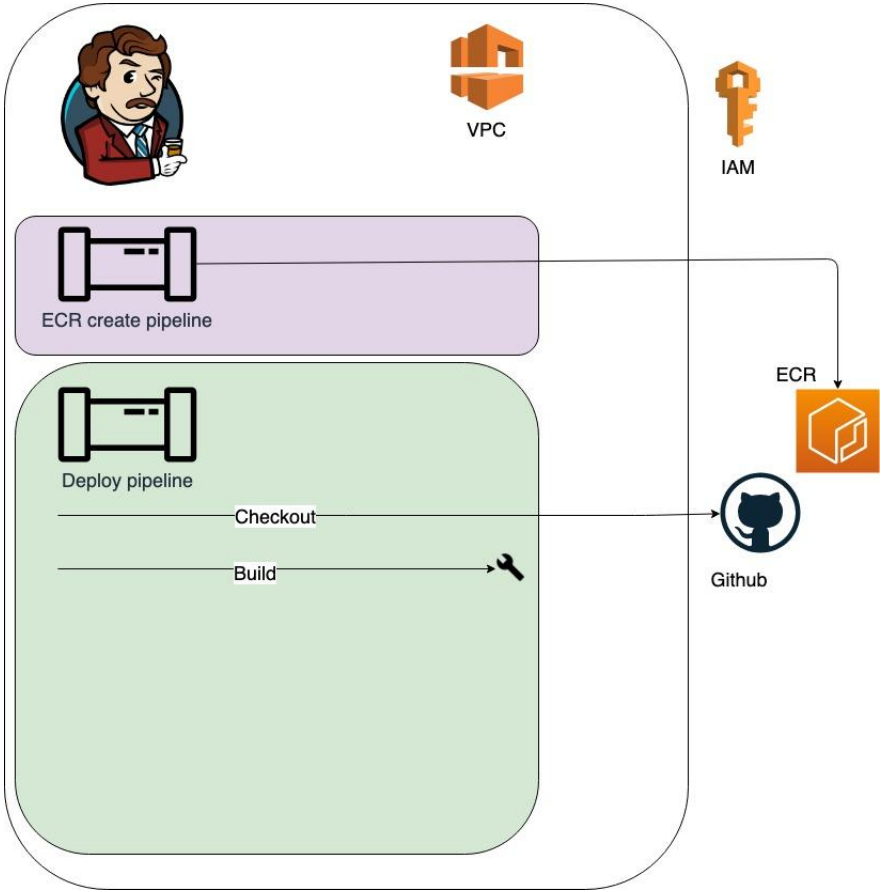
JENKINS



JWORKS
POWERED BY ORDINA

ORDINA

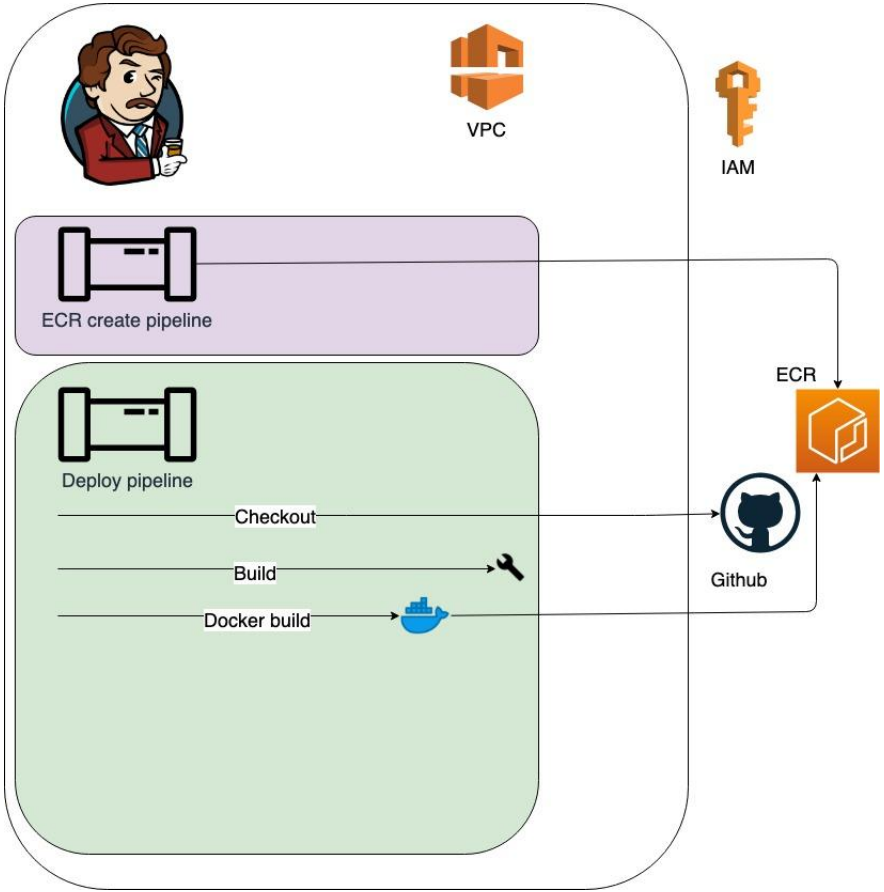
JENKINS



JWORKS
POWERED BY ORDINA

ORDINA

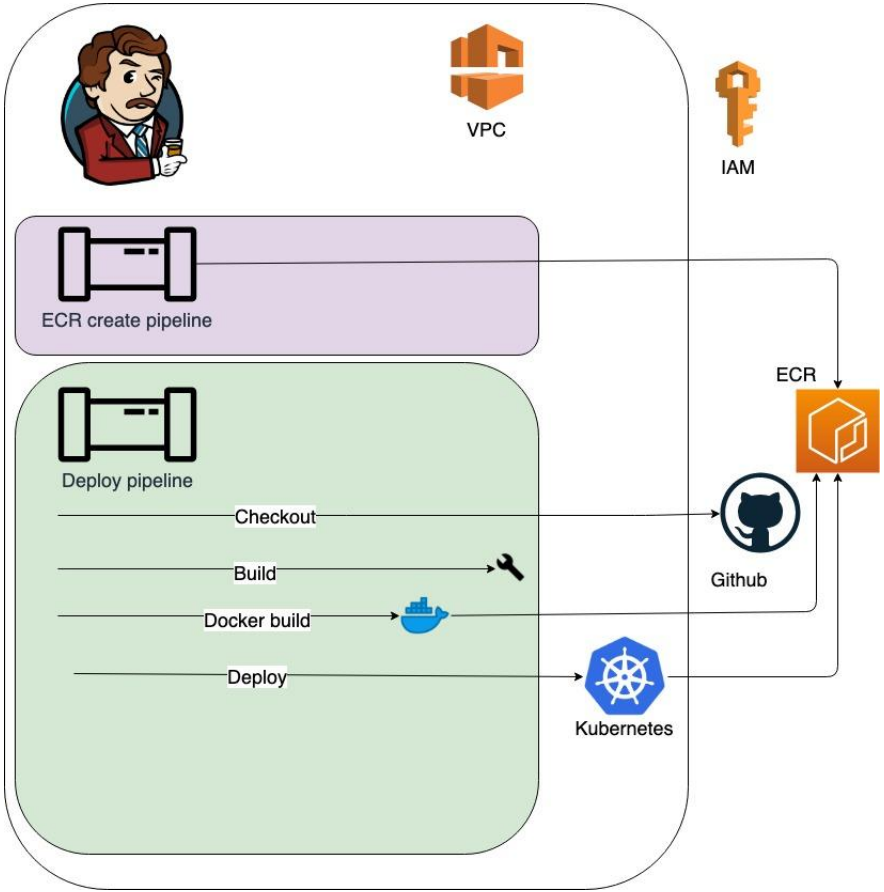
JENKINS



JWORKS
POWERED BY ORDINA

ORDINA

JENKINS



WORKSHOP TIME

WORKSHOP

Prerequisites

- Python (3 preferred) & pip
  - Mac: brew install python3 python3-pip
  - Windows: https://www.python.org/downloads/windows/
- Kubectl
  - https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html
- aws-cli (needs Python 3 and pip)
  - https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html
- aws-iam-authenticator
  - https://docs.aws.amazon.com/eks/latest/userguide/install-aws-iam-authenticator.html

TOOLS & TIPS

Tools to improve your productivity

- Kubectx
  - Easily switch between clusters and Kubernetes providers
  - https://github.com/ahmetb/kubectx/
- Kubens
  - Easily switch between namespaces
  - https://github.com/ahmetb/kubectx/blob/master/kubens

WORKSHOP

- Teams of 2
- Basic Spring Boot TODO application
  - Every team has his own branch
  - Clone the repo with your branch: git clone -b <branch name> git@github.com:ordina-jworks/k8s-demo.git
  - Please ask for access if you don't have access to the repository
- Create a Dockerfile which will create a Docker image of our application
- Create a YAML file that handles the deployment of the application on Kubernetes
- Setup a Jenkins pipeline to build and deploy your image on the Kubernetes cluster
- At the end of the workshop, your application should be available on the cloud with the help of an automatic pipeline

How do I ‘dockerize’ my Spring Boot application?

- Use JRE, not JDK
  - You don’t need the full JDK to run a Java application!
- Use alpine image if you can
  - Very lightweight
  - Only contains the basics

How do I ‘dockerize’ my Spring Boot application?

- Docker build & run (for testing purposes)
  - docker build -t k8s-demo .
  - docker run -p 8080:8080 k8s-demo
    - Open port 8080 from your Docker container to your machine
  - Environment variables
    - docker run -p 8080:8080 -e “WELCOME_NAME=JWorks” k8s-demo

Deploying my Docker image to a Kubernetes cluster

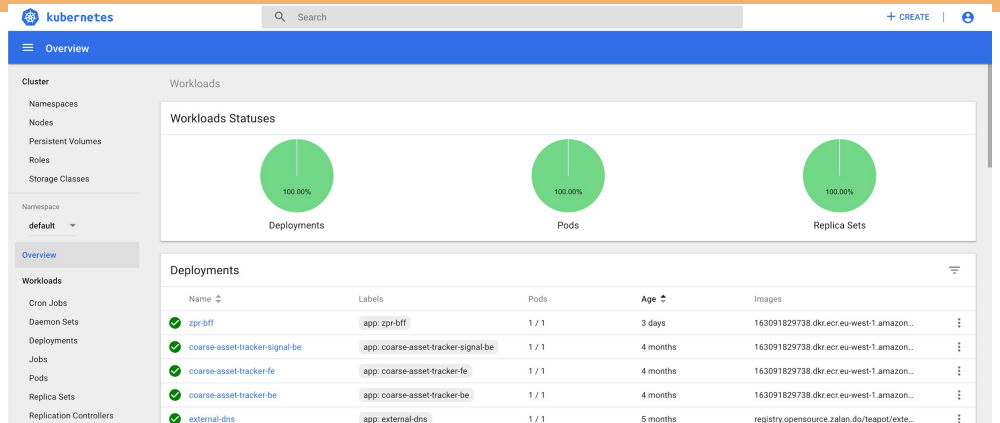
- Set your base path in application-prod.properties for Thymeleaf support
  - ex. for team 3: k8s-demo-team3
- Recreate our Docker image
  - Need to create a Docker repository first!
  - New docker image with new name and tag from our Docker repository
  - Authenticate with ECR
    - aws ecr get-login --region eu-west-1 --no-include-email
- Kubernetes uses YAML files
- Enable production mode in Spring Boot in YAML
  - name: SPRING_PROFILES_ACTIVE
  - value: prod
- Ingress

WORKSHOP

Kubernetes Dashboard

- Execute the following command in the Terminal:
  kubectl proxy
- Get token:
  - http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/#!/login
- Then navigate to:

http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/#!/login



JWORKS
POWERED BY ORDINA

ORDINA

WORKSHOP

Jenkins pipeline

- Automatic pipeline which will deploy your application
- Use environment variables from Jenkins
  - Image name: <project_name>:$BUILD-NUMBER-$GIT_COMMIT
  - use envsubst pipe for YAML files
    - YAML can’t process environment variables
    - envsubst replaces environment variables with values

```
envsubst < k8s/demo-deployment.yaml | kubectl apply -f -
```

TRY THIS AT HOME

Want to set up your own Kubernetes cluster to play with?

- Various free K8S providers for testing purposes
  - https://tryk8s.com/
  - https://kubesail.com/
  - Minikube - A local Kubernetes cluster for testing purposes only!
  - Free signup credits from AWS / GCP
  - MicroK8s
  - Can also use the AWS JWorks account!
- Learn more about AWS / EKS:
  - https://www.aws.training/
  - https://eksworkshop.com/
  - JWorks Docs: https://jworks.cfapps.io/
- Learn more about Kubernetes:
  - Kubernetes official documentation: https://kubernetes.io/
  - Blogpost: https://ordina-jworks.github.io/cloud/2019/08/05/deploy-spring-boot-kubernetes.html